



**Centro Universitário de Brasília
Instituto CEUB de Pesquisa e Desenvolvimento - ICPD**

JEAN CARLO GALDINO RODRIGUES

**PROPOSTA DE UM PROCESSO DE GERÊNCIA DE MUDANÇAS
PARA PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE
ITERATIVO E INCREMENTAL ALINHADO À GOVERNANÇA DE TI
EM CONFORMIDADE COM O FRAMEWORK COBIT**

Brasília
2013

JEAN CARLO GALDINO RODRIGUES

**PROPOSTA DE UM PROCESSO DE GERÊNCIA DE MUDANÇAS
PARA PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE
ITERATIVO E INCREMENTAL ALINHADO À GOVERNANÇA DE TI
EM CONFORMIDADE COM O FRAMEWORK COBIT**

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB/ICPD) como pré-requisito para obtenção de Certificado de Conclusão de Curso de Pós-graduação *Lato Sensu* em Governança de TI

Orientador: Fabiano Mariath

Brasília
2013

JEAN CARLO GALDINO RODRIGUES

**PROPOSTA DE UM PROCESSO DE GERÊNCIA DE MUDANÇAS
PARA PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE
ITERATIVO E INCREMENTAL ALINHADO À GOVERNANÇA DE TI
EM CONFORMIDADE COM O FRAMEWORK COBIT**

Trabalho apresentado ao Centro
Universitário de Brasília (UniCEUB/ICPD)
como pré-requisito para a obtenção de
Certificado de Conclusão de Curso de
Pós-graduação *Lato Sensu* em
Governança de TI

Orientador: Prof. Fabiano Mariath

Brasília, ____ de _____ de 2013.

Banca Examinadora

Prof. Dr. Nome completo

Prof. Dr. Nome completo

AGRADECIMENTOS

Agradeço a Deus, à minha família, amigos, e também ao meu orientador Fabiano Mariath que me deu apoio neste trabalho.

RESUMO

Este estudo faz uma proposta de um processo de Gerência de Mudanças para processos de desenvolvimento de software iterativo e incremental alinhado à Governança de TI em conformidade com o framework CobIT. Para isso, este trabalho descreve a importância do processo de Gerência de Mudanças na construção e manutenção de sistemas em uma organização, mostrando que este processo evita uma maior quantidade de retrabalho e de atrasos na entrega de um sistema. Esta pesquisa descreve também como surge uma necessidade de mudança e como ela impacta no desenvolvimento de sistemas. Foi desenvolvido neste estudo uma proposta para formalização do processo de Gerência de Mudanças como forma de padronizar as solicitações e registros de mudanças em sistemas computacionais. Para isso, foi feita uma correlação do processo de desenvolvimento de software Open Up, já consagrado no mercado, com os objetivos de controle do processo de Gerência de Mudanças do framework CobIT. Conclui-se então esse estudo com a proposta formal de Gerenciamento de Mudanças para o modelo de desenvolvimento de sistemas Open Up em atendimento aos controles do CobIT.

Palavras-chave: Gerência de Mudança. Governança de TI. Desenvolvimento de Software Iterativo. Open UP. Plugin EPF

ABSTRACT

This research proposes a Change Management process for Iterative and Incremental Software Development process aligned to IT Governance in accordance with CobIT framework. In this regard, this work describes the importance of the Change Management Process on building and maintenance of systems in an organization, showing that this process avoids a large amount of rework and delays in delivering systems. This research also demonstrates how a necessity of change emerges and how it impacts on systems development. In this work, it was developed a proposal to formalize the Change Management process as a way to standardize the registration and requests of change for computing systems. Therefore, it has done a correlation of Open Up software development process, which is already consecrated in the market, with the Control Objectives of the CobIT's Change Management process. It was concluded in this study the formal proposal for Change Management into Open Up systems development model in accordance with the CobIT's Control Objectives.

Key words: Change management. IT Governance. Iterative Software Development. Open Up. Plugin EPF

LISTA DE FIGURAS

Figura 1 – Estatística de sucesso em projetos de software do Extreme Chaos	10
Figura 2 - TI e Sistema de Informação num contexto de negócio	14
Figura 3 - Custo de Mudança no Projeto no Modelo em Cascata	16
Figura 4 -Custo de mudanças no projeto com metodologias ágeis.....	19
Figura 5- Etapas da Mudança Organizacional.	21
Figura 7 - Áreas-Foco da Governança de TI, na visão do CobiT	23
Figura 8 Domínios do CobiT no Framework.....	25
Figura 9. Tabela RACI.....	27
Figura 10 Processo Formal do Fluxo de Gerência de Mudanças.....	35

LISTA DE QUADROS

Quadro 1 - Comparativo entre o processo convencional e a abordagem do modelo iterativo de desenvolvimento de software	17
Quadro 2 – Comparação de papéis Open UP x CobIT	33
Quadro 3 – Relação dos Objetivos de Controle do CobIT com Atividades de Gerência de Mudança	36
Quadro 4 - Relação de Atividades x Papéis	37

SUMÁRIO

INTRODUÇÃO	9
1 A INFLUÊNCIA DA GERÊNCIA DE MUDANÇAS NA CONSTRUÇÃO DE SOFTWARES	13
1.1 O ELEMENTO SOFTWARE NO SISTEMA DE INFORMAÇÃO	13
1.2 MODELO ITERATIVO DE CONSTRUÇÃO DE SISTEMAS DE INFORMAÇÃO	15
1.3 MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE	18
1.4 O MODELO DE PROCESSO UNIFICADO COM OPEN UP	19
2 GESTÃO DE MUDANÇAS SEGUNDO A ÓTICA DO COBIT 4.1	21
2.1 PROCESSO DE GERÊNCIA DE MUDANÇAS	26
3 PROCESSO E SUAS FORMAS DE PUBLICAÇÃO	28
3.1 ECLIPSE PROCESS FRAMEWORK COMPOSER	29
4 DESENVOLVIMENTO	31
4.1 PROPOSTA DE UM MODELO EM FORMA DE PLUGIN DE GERÊNCIA DE MUDANÇAS EM ATENDIMENTO AO COBIT	32
CONCLUSÃO	40
REFERÊNCIAS	41
APÊNDICE A – CRIAÇÃO DO PLUGIN GERÊNCIA DE MUDANÇAS NO EPF PARA O OPEN UP	43

INTRODUÇÃO

Na modernidade de hoje, as empresas e organizações têm sido alvo das tão discutidas mudanças organizacionais. No entanto, as mudanças ocorrem inevitavelmente no cotidiano sobre qualquer aspecto, não se restringindo ao contexto organizacional. É indiscutível o fato de que várias transformações tenham afetado profundamente a sociedade, influenciando intensivamente para a evolução mundial. Concorrentemente, a evolução da computação e dos sistemas de informação está cada vez mais oferecendo novas soluções em automatização e otimização de processos às empresas e órgãos públicos.

O ITGI (2007) cita que cada vez mais a Alta Direção vem percebendo o grande impacto que a informação possui no sucesso da organização. Portanto, os executivos almejam um alto entendimento sobre a forma como a TI funciona e o quanto ela está sendo bem administrada para alcançar vantagens competitivas.

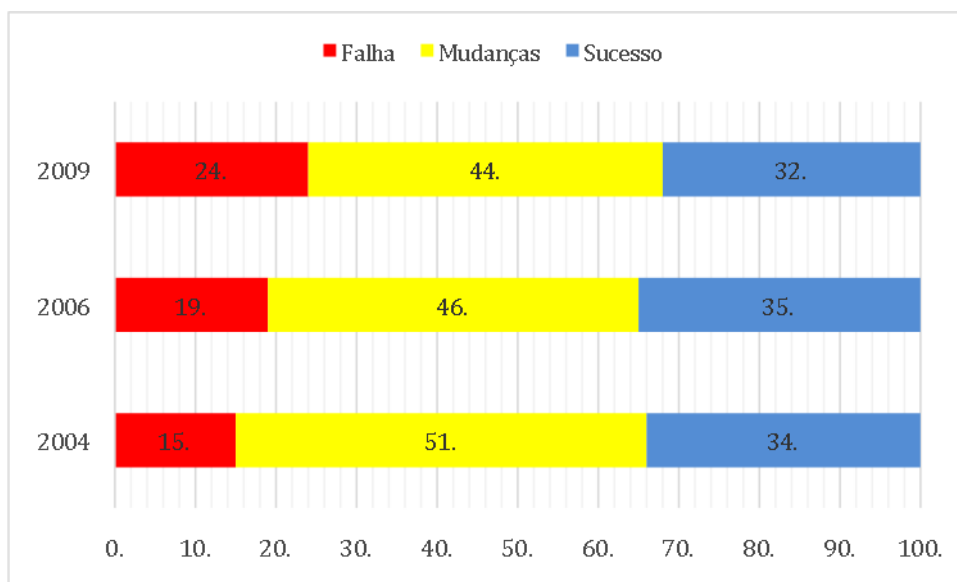
Para essas organizações tornarem-se mais competitivas, buscam cada vez mais investir em aquisição e desenvolvimento de sistemas de informação para atenderem às suas expectativas no gerenciamento da informação.

Segundo o ITGI (2007) “organizações bem-sucedidas reconhecem os benefícios da tecnologia da informação e a utiliza para direcionar os valores das partes interessadas no negócio” e também complementa com “a necessidade da avaliação do valor de TI, o gerenciamento dos riscos relacionados à TI e as crescentes necessidades de controle sobre as informações são agora entendidos como elementos-chave da governança corporativa”. Estes aspectos são considerados no framework CobIT.

No entanto, ainda segundo o ITGI (2007), devido à complexidade na construção de sistemas, alguns sistemas tendem a passar por retrabalho e defeitos na entrega de soluções e serviços por não satisfazerem alguns requisitos necessários para que os sistemas sejam de alta confiabilidade.

De acordo com a pesquisa realizada em 2009 pelo Standish Group, nos Estados Unidos, que avalia milhares de projetos de software envolvendo mais de 300 organizações, somente 32% dos projetos obtiveram sucesso considerando os critério de prazo, orçamento e escopo. Conforme ilustrado na Figura 1, 44% dos projetos mudaram e isso gerou atrasos, estouros de orçamento, e/ou redução de escopo e 24% dos projetos falharam sendo cancelados durante seu desenvolvimento ou entregues, mas nunca utilizados.

Figura 1 – Estatística de sucesso em projetos de software do Extreme Chaos



Fonte: Gráfico adaptado de Standish Group (2009)

Segundo esse estudo, um dos maiores fatores para aumento de custo e de prazos são os constantes retrabalhos.

Desta forma, um dos processos mais importantes em um desenvolvimento de um sistema que busca evitar as falhas e defeitos na entrega é a Gerência da Mudança que é onde se planeja e se controla as solicitações de mudanças nos sistemas com o objetivo de diminuir as paradas de um sistema e o retrabalho causado por especificações de mudança inadequadas.

Conforme destaca Senge (2000), o sucesso da organização dependerá do grau em que ela conseguirá sustentar seus processos de mudança e transformação.

Uma solicitação de mudança pode originar por vários fatores. Segundo Chiavenato (2003), a administração da mudança inicia-se com a análise das forças ambientais e internas que levam a criação da necessidade de mudanças em uma organização, devendo sempre atentar aos problemas e oportunidades.

Para Chiavenato (2008), o agente de mudança pode ser uma pessoa, um grupo, uma organização ou mesmo a sociedade, podendo estes provocarem vários tipos de mudanças organizacionais.

Entre os diversos processos do framework CobIT está o de Gerencia de Mudanças. Processo este que visa a obter formalização e padronização dos pedidos de mudanças que podem decorrer em uma solução de TI.

Essa pesquisa busca fornecer um levantamento de informações da área de gerenciamento de mudança e sua aplicação nas etapas de desenvolvimento de sistemas, a fim de reduzir as falhas na implantação de mudanças em uma solução de TI. Essa pesquisa inclui a elaboração de um plug-in de processo no Eclipse Process Framework baseado no Open UP para atender o processo de Gerenciar Mudanças do CobiT.

O objetivo deste trabalho é apresentar o desenvolvimento da criação de um mecanismo baseado no desenvolvimento de software iterativo Open UP que esteja em conformidade com os objetivos de controle do COBIT no que se refere à gestão de mudanças AI06

O presente trabalho foi então estruturado em quatro capítulos.

No capítulo primeiro, apresenta-se a conceituação de sistemas de informação e seu desenvolvimento, o segundo capítulo proporciona uma análise sobre governança de TI com o COBIT e seu processo de Gerencia de Mudanças; no terceiro capítulo mostra-se a ferramenta de modelagem de processos Eclipse Process Framework Composer (EPF) e no quarto capítulo encontra-se o desenvolvimento da criação de um mecanismo baseado no desenvolvimento de software iterativo que esteja em conformidade com os objetivos de controle do COBIT no que se refere à gestão de mudanças AI06.

1 A INFLUÊNCIA DA GERÊNCIA DE MUDANÇAS NA CONSTRUÇÃO DE SOFTWARES

Este capítulo descreve os elementos teóricos envolvidos no processo de Gerência de Mudanças de um Desenvolvimento de Sistemas assim como também o framework de Governança de TI

1.1 O ELEMENTO SOFTWARE NO SISTEMA DE INFORMAÇÃO

A definição de sistema de informação segundo Pressman (2007) é “um conjunto ou disposição de elementos que é organizado para executar certo método, procedimento ou controle ao processar informações”.

Assim, o sistema é constituído de vários elementos como citado abaixo:

- Software: que são programas de computador, estruturas de dados e documentação correlata que servem para efetivar o método, processo ou controle lógico necessário.
- Hardware: dispositivos eletrônicos que fornecem capacidade ao computador, e dispositivos eletromecânicos que oferecem funções ao mundo externo.
- Pessoas: Usuários e operadores de hardware e software.
- Banco de dados: Uma grande e organizada coleção de informações a que se tem acesso pelo software e faz parte integrante da função do sistema.
- Documentação: Manuais, formulários e outras informações descritivas que retratam o uso e/ou operação do sistema.
- Procedimentos: Os passos que definem o uso específico de cada elemento do sistema ou o contexto processual em que o sistema reside. (PRESSMAN, 2007, p. 179)

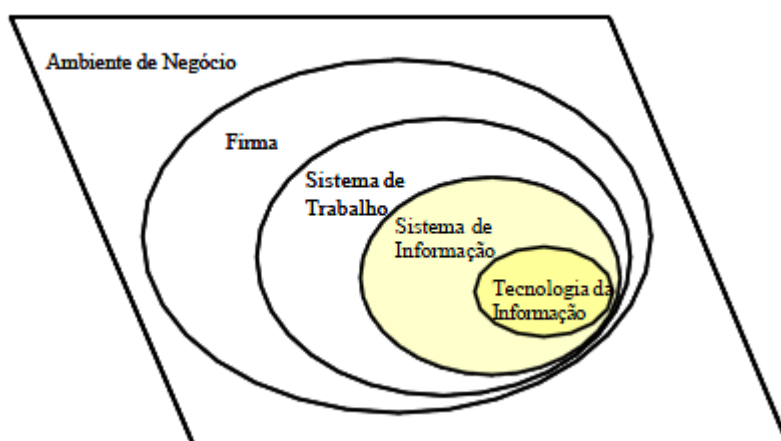
Já para O'Brien (2004), sistema de informação é um conjunto de pessoas, software, hardware, redes de comunicações recursos de dados que são coletados, transformados e que disseminam informações nas organizações. Podendo ser definido como um conjunto de elementos inter-relacionados que formam as características das empresas.

Na visão de Oliveira (2007), sistema de informação é o processo de transformação de dados em informações. Quando esse processo está voltado para a geração de informações que são necessárias e utilizadas no processo decisório da organização, diz-se trata-se de um sistema de informações

gerenciais. Verifica-se ainda que o processo administrativo apresenta a tomada de decisões como elemento básico; e para processo decisório adequado, é necessário ter um sistema de informação eficiente.

Uma visão de como os sistemas de informação estão inseridos nas organizações pode ser vista na figura 2. De acordo com este modelo Alter (1999), há um sistema de trabalho em que os participantes desempenham um processo de negócio utilizando informação, tecnologia, e outros recursos para produzir produtos para clientes internos e externos. O termo trabalho tem o significado de aplicação de recursos humanos e físicos, tal como pessoas, equipamentos, tempo, esforço e capital para gerar os produtos.

Figura 2 - TI e Sistema de Informação num contexto de negócio



Fonte: Alter (1999, p. 43)

De acordo com Boehm (2006) em seu artigo, por volta de 1950, o desenvolvimento de software era realizado da mesma forma que a engenharia de hardware em que se baseava em fazer várias análises antes de começar a codificar. Mais tarde em 1960, havia uma aproximação maior com a codificação de software mas gerando códigos confusos e mal estruturados. Assim em 1970, utilizaram-se do método cascata para desenvolvimento de sistemas, em que se une os benefícios dos dois modelos utilizados nas décadas passadas, surgindo então métodos formais e métodos estruturais de codificação de sistemas. Em 1980, utilizavam a reusabilidade através de métodos orientados a objetos, aumentando assim a produtividade das empresas no desenvolvimento de softwares. Nesta época também

surgiram modelos de maturidade e padronizações assim como também fábrica de softwares. No início de 1990, a importância da rapidez na entrega de softwares e o surgimento do UML, levou ao abandono do modelo cascata e ao surgimento de um novo modelo de desenvolvimento de software baseado em processos concorrentes orientado a usuário. A partir de 2000, surgiram os métodos ágeis que procuravam um desenvolvimento com iterações menores, desenvolvimento rápido e com menos burocracia com documentação.

1.2 MODELO ITERATIVO DE CONSTRUÇÃO DE SISTEMAS DE INFORMAÇÃO

De acordo com Whitten (1995), várias organizações que desenvolvem software não possuem uma visão clara de como deveria ser um processo definido, reproduzível, previsível de desenvolvimento e que um processo disciplinado leva a um aumento expressivo do risco em prever e controlar fatores cruciais tais como custo, duração e qualidade. Assim, um processo pode ser definido como uma abordagem sistemática que é executada para atingir um objetivo específico. Dessa forma, no âmbito de desenvolvimento de software, um processo é definido como um conjunto ordenado de atividades que, após suas conclusões, resultam em um produto a ser entregue ao cliente.

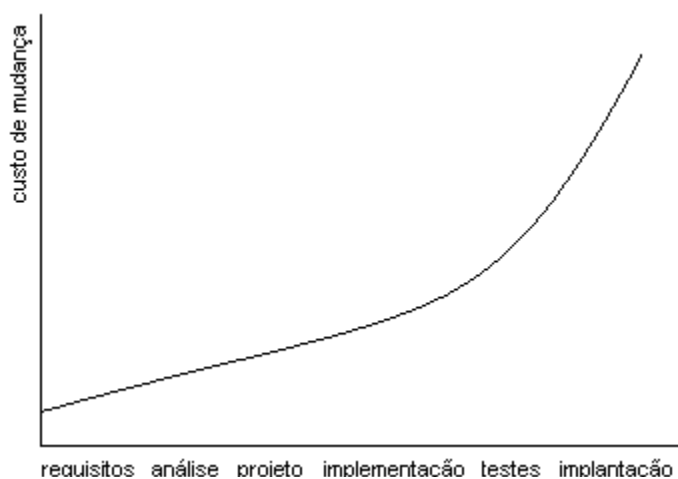
Existem diversas metodologias de desenvolvimento de sistemas, entre elas a Cascata, que é uma forma tradicional e sobrecarregada. Segundo Pressman (2001), essa metodologia é composta basicamente por atividades sequenciais de levantamento de requisitos, análise, projeto, implementação, teste, implantação e manutenção. Este modelo deriva de outras engenharias tradicionais (Civil, Elétrica, Naval,...) e foi o primeiro a ser usado pela Engenharia de Software, na década de 70.

O modelo Cascata foi a forma predominante de desenvolvimento de software até o início dos anos 90, apesar das críticas de pesquisadores e desenvolvedores da área que identificaram os problemas relacionados ao se adotar essa visão sequencial de tarefas. Corroborando essa visão, Brooks (1987) em seu artigo “No Silver Bullet: Essence and Accidents of Software Engineering”, descreve

que a ideia de especificar totalmente um software antes do início de sua implementação é impossível.

Segundo Royce (1998), a estrutura básica do modelo cascata é propensa a falhas por ter a fase de teste ocorrendo no final do ciclo de desenvolvimento que trata de experimentar o que foi analisado e codificado, portanto as mudanças de projeto a partir desta fase são provavelmente difíceis de serem implementadas, assim como mudanças nos requisitos iniciais.

Figura 3 - Custo de Mudança no Projeto no Modelo em Cascata



Fonte: (Schwaber & Beedle, 2002)

Royce em seu estudo, defende um modelo iterativo, com divisão nas seguintes etapas:

- Estágio de Engenharia
- Concepção
- Elaboração
- Estágio de produção
- Construção
- Transição

Este modelo é baseado em 10 princípios que suportam essa nova abordagem:

- I. O processo deve ser focado em obter o mais rápido possível a arquitetura do sistema;
- II. Estabelecer um processo iterativo de ciclo de vida, que procure reduzir o risco a cada iteração;
- III. Os métodos de desenvolvimento devem ser adaptados de forma a permitir um desenvolvimento baseado em componentes;

- IV. Estabelecer um ambiente de gerenciamento de mudança;
- V. Utilizar procedimentos e ferramentas que permitam automatizar o processo de desenvolvimento
- VI. Capturar e registrar a arquitetura e os demais artefatos de desenvolvimento em uma notação baseada em modelos;
- VII. Utilizar uma abordagem baseada em demonstrações dos produtos intermediários do projeto para avaliar os modelos, de forma a melhorar o entendimento e eliminar defeitos de arquitetura ou de projeto o quanto antes;
- VIII. Planejar liberações de produtos intermediários para serem testados em grupos de diferentes cenários de funcionamento, aumentando o nível de detalhes a cada novo teste;
- IX. Definir métricas e instrumentos para avaliar objetivamente a qualidade do produto sendo desenvolvido;
- X. Estabelecer um processo que seja configurável de acordo com as necessidades de cada aplicação/projeto, porém mantendo a linha básica de padrões de controle. (Royce, 1998)

No quadro 1, são apresentadas comparações entre o processo tradicional de desenvolvimento de software e a nova abordagem proposta pelo modelo iterativo:

Quadro 1 - Comparativo entre o processo convencional e a abordagem do modelo iterativo de desenvolvimento de software

Processo convencional (10 principais riscos)	Impactos	Processo Iterativo (resolução intrínseca do risco)
1) Identificação tardia de mudanças, retrabalho excessivo	Qualidade, Custo, Prazo	<ul style="list-style-type: none"> • Abordagem de foco na arquitetura inicial • Desenvolvimento Iterativo • Gerenciamento da mudança • Processo de identificação e tratamento de riscos
2) Conflitos entre membros da equipe	Qualidade, Custo, Prazo	<ul style="list-style-type: none"> • Iterações incentivadas desde o início do projeto para evitar conflitos de entendimento • Planejamento e gerenciamento confiáveis
3) Recursos de desenvolvimento inadequados	Custo, Prazo	<ul style="list-style-type: none"> • Ambientes com modelos de desenvolvimento modernos e eficientes • Ambiente de desenvolvimento integrado • Metodologia de desenvolvimento com modelos integrando todas as fases do ciclo de vida • Uso de ferramentas com apoio às metodologias
4) Conflitos e choques de interesses entre os envolvidos no projeto (stakeholders)	Custo, Prazo	<ul style="list-style-type: none"> • Revisões de projeto baseadas em demonstrações • Requisitos e testes baseados em casos
5) Inserção de tecnologia necessária	Custo,	<ul style="list-style-type: none"> • Abordagem focada na arquitetura inicial

	Prazo	<ul style="list-style-type: none"> • Desenvolvimento baseado em componentes
6) Lentidão e falta de objetividade no levantamento de requisitos	Custo, Prazo	<ul style="list-style-type: none"> • Desenvolvimento iterativo • Modelagem baseada em casos • Revisão baseada em demonstrações
7) Interrupções na fase análise	Prazo	<ul style="list-style-type: none"> • Revisão baseada em demonstrações • Requisitos e testes baseados em casos
8) Desempenho inadequado da aplicação	Qualidade	<ul style="list-style-type: none"> • Avaliação de desempenho baseada em demonstrações • Verificação antecipada do desempenho da arquitetura
9) Ênfase excessiva em documentações em forma de texto e burocracia de aprovação	Prazo	<ul style="list-style-type: none"> • Avaliação baseada em demonstrações • Controle objetivo de qualidade
10) Desenvolvimento de funcionalidades inadequadas	Qualidade	<ul style="list-style-type: none"> • Desenvolvimento iterativo • Desenvolvimento de protótipos, liberações incrementais

Fonte: Royce (1998)

1.3 MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE

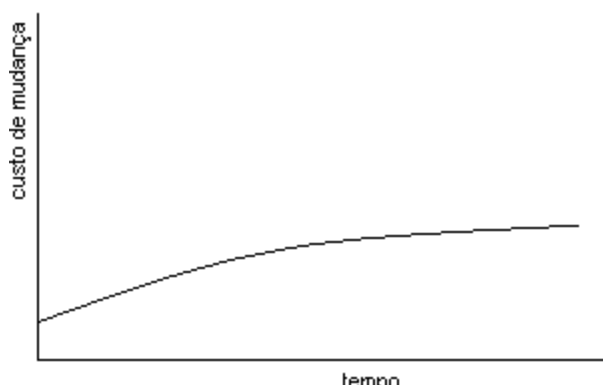
Para atender as crescentes necessidades do negócio de hoje um novo modelo foi introduzido para o desenvolvimento de software no final dos anos 80: o método ágil.

De acordo com Szalvay (2004), esse método foi introduzido com consciência da visão das limitações existentes dos métodos tradicionais. Esse método adota o modo incremental e iterativo para melhorar a produtividade e a eficiência de um processo de desenvolvimento de software.

Segundo o Agile Manifesto (2004) os conceitos chaves do manifesto ágil são:

- **Indivíduos e interações** ao invés de processos e ferramentas.
- **Software executável** ao invés de documentação.
- **Colaboração do cliente** ao invés de negociação de contratos.
- **Respostas rápidas a mudanças** ao invés de seguir planos.

Figura 4 -Custo de mudanças no projeto com metodologias ágeis



Fonte: Schwaber & Beedle(2002)

No gráfico fica claro que o custos de mudanças ocorridas utilizando metodologias ágeis não tendem a crescer tanto quanto na metodologia em cascata conforme apresentado na figura 5.

1.4 O MODELO DE PROCESSO UNIFICADO COM OPEN UP

Segundo Balduino (2007), projetos diferentes possuem necessidades diferentes de processos, sendo que fatores típicos levam a escolha para um processo formal ou ágil, podendo estes fatores serem como o tamanho da equipe, complexidade da arquitetura, novas tecnologias, etc.

Ainda segundo este autor, o OpenUp é um processo de desenvolvimento ágil de software onde apenas o conteúdo essencial é incluído e ainda há a possibilidade de ser estendido para atender outras necessidades.

Este processo possui os seguintes princípios:

- **Balancear as prioridades concorrentes para maximizar os valores dos Stakeholders** - Promover práticas que permitam à equipe de desenvolvimento e aos Stakeholders desenvolver uma solução que contrabalanceie todas as necessidades dos Stakeholders e que esteja de acordo com as restrições propostas no projeto;
- **Colaborar para alinhar os interesses e compartilhar os conhecimentos** - Promover as práticas que promovam um ambiente saudável de desenvolvimento em equipe, possibilitando a colaboração e

possibilitando a compreensão e concordância sobre os principais requisitos que definem o sistema.

- **Focar inicialmente na arquitetura para minimizar riscos e organizar o desenvolvimento**□- Promover as práticas que permitam à equipe de desenvolvimento focar suas ações na arquitetura, buscando minimizar os riscos e organizar o processo de desenvolvimento da solução proposta.

- **Envolver os Stakeholders para obter contínuo feedback do desenvolvimento**□- Promover práticas que permitam à equipe de desenvolvimento obter feedback contínuo dos stakeholders sobre a solução proposta e demonstrar o incremento de seu valor.

Assim, Balduino (2007) cita as disciplinas do Open Up :

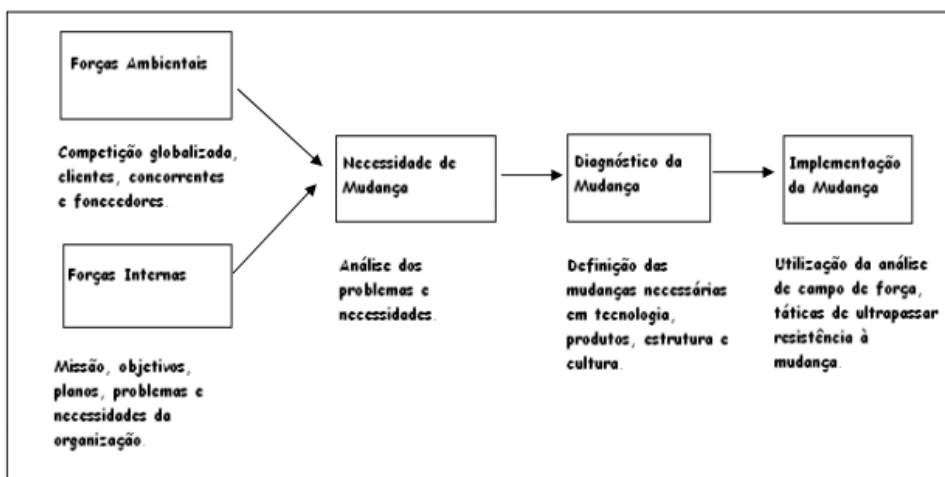
- Requisitos
- Arquitetura
- Desenvolvimento
- Teste
- Gerencia de Projetos
- Gerencia de Configuração e de Mudança.

No Open UP, existem 18 tarefas que podem ser executados por papéis primários (responsáveis pela execução da tarefa) ou por executores adicionais (apoando e provendo informação usada para a execução da tarefa).

2 GESTÃO DE MUDANÇAS SEGUNDO A ÓTICA DO COBIT 4.1

Uma solicitação de mudança pode originar por vários fatores. Segundo Chiavenato (2003), a administração da mudança inicia-se com a análise das forças ambientais e internas que levam a criação da necessidade de mudanças em uma organização, devendo sempre atentar aos problemas e oportunidades, pois a necessidade percebida é que permite o diagnóstico da mudança, para a implantação de forma planejada e organizada, conforme ilustra na figura abaixo.

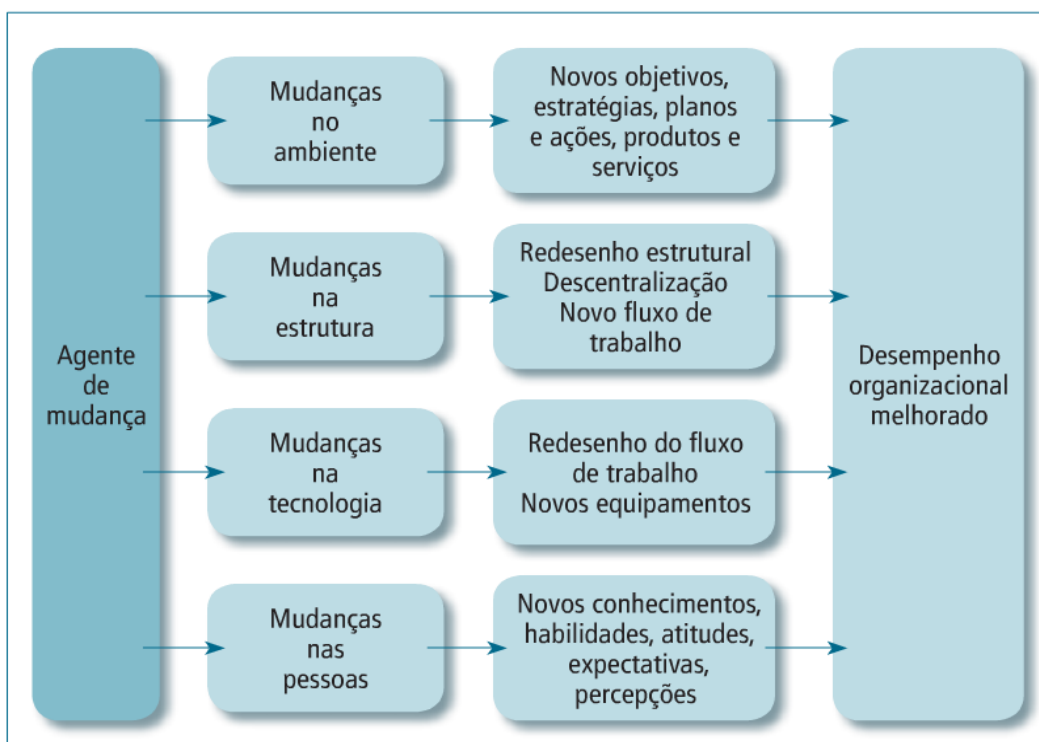
Figura 5- Etapas da Mudança Organizacional.



Fonte: Chiavenato (2003)

Para Chiavenato (2008), o agente de mudança pode ser uma pessoa, um grupo, uma organização ou mesmo a sociedade, podendo estes provocarem vários tipos de mudanças organizacionais, conforme representados na figura abaixo:

Figura 6 – Os vários tipos de mudanças organizacionais nas empresas



Fonte: Chiavenato(2008)

Segundo Heldman (2006) as mudanças surgem por vários motivos e não necessariamente implicam em consequências negativas, mas como também gerar resultados positivos. Esse processo é de grande importância porque mudanças em excesso ou mesmo uma mudança significativa poderá afetar o custo, o cronograma, o escopo e/ou a qualidade de um projeto de sistema.

Assim, na área de software para Heldman (2006), a mudança ocorre à medida que o projeto de sistema evolui, seja por uma solicitação de um dos envolvidos na construção de um sistema, ou durante as atividades em que um dos membros envolvidos descobrem métodos mais eficientes de realizar tarefas, e pode também a mudança ocorrer em decorrência de erros cometidos nas etapas iniciais do projeto, ou por ser resultado de planos de contingência.

No entanto, as demandas para alterações descontroladas em sistemas levam rapidamente as soluções de TI ao caos. Assim, segundo o Pressman (2007) deve-se combinar procedimentos humanos e ferramentas automatizadas para proporcionar um mecanismo de controle de mudanças.

Organizações bem sucedidas reconhecem os benefícios da tecnologia da informação e a utiliza para direcionar os valores das partes interessadas no negócio.

Assim, segundo o ITGI (2007) o uso da Governança de TI surge-se com a necessidade da avaliação do valor da TI, de seus riscos e a crescente demanda sobre o controle de informações em que constituem como elementos-chave da governança corporativa.

Esse modelo, de acordo com Fernandes (2008) origina da abreviação de Control Objectives for Information and related Technology e foi criado em 1994 pela ISACF. Trata-se de um modelo que vem evoluindo constantemente, surgindo novas versões deste modelo. Seu objetivo é a contribuição para entregar produtos e serviços de TI com sucesso, utilizando da visão das necessidades do negócio, focando mais no controle do que na execução.

Para esse modelo, há cinco áreas consideradas como pilares fundamentais que sustentam o núcleo da Governança de TI, conforme figura 1.

Figura 7 - Áreas-Foco da Governança de TI, na visão do CobiT



Fonte: ITGI (2007)

Segundo esse framework, o sucesso da área de TI na entrega dos serviços requeridos pelo negócio será atingido com a implementação pelos executivos de um sistema interno de controles ou uma metodologia. O CobiT contribui para isso ao:

- *Fazer uma ligação com os requisitos de negócios.*
- *Organizar as atividades de TI em um modelo de processos geralmente aceito.*
- *Identificar os mais importantes recursos de TI a serem utilizados.*
- *Definir os objetivos de controle gerenciais a serem considerados.* (ITGI, 2007)

Assim, o CobiT utilizando-se do ciclo de melhoria contínua(planejar, construir, executar, monitorar), esse modelo identificou 34 processos de TI distribuídos entre quatro domínios, que mostram os agrupamentos comuns existentes em uma organização padrão de TI. (FERNANDES, 2008, p. 178). Esses domínios são:

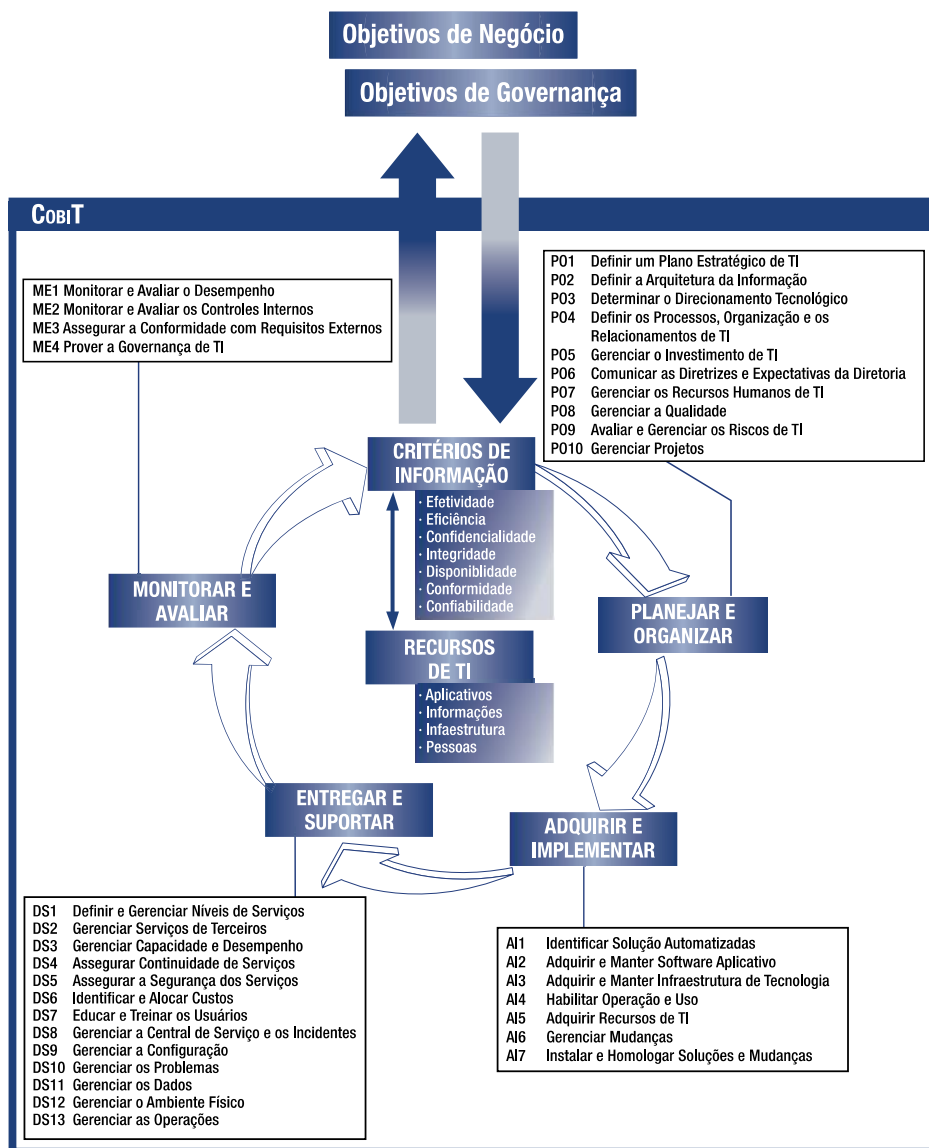
- *Planejar e Organizar(PO) - Provê direção para entrega de soluções (AI) e entrega de serviços (DS)*
- *Adquirir e Implementar(AI)- Provê as soluções e as transfere para tornarem-se serviços*
- *Entregar e Suportar(DS)- Recebe as soluções e as torna passíveis de uso pelos usuários finais.*
- *Monitorar e Avaliar(ME)- Monitora todos os processos para garantir que a direção definida seja seguida.* (ITGI, 2007)

O foco desta pesquisa está centralizado no domínio Adquirir e Implementar(AI) por possuir o processo de Gerência de Mudanças(AI6) que será mais detalhado posteriormente no capítulo 2.1.

Assim, o domínio de Adquirir e Implementar é onde identificam-se , desenvolvem-se ou adquirem as soluções de TI. E ainda, esse domínio cobre as alterações e manutenções nos sistemas existentes para assegurar que as soluções atendem aos objetivos de negócios continuamente. O CobiT Framework ITGI (2007) trata das questões de gerenciamento como as seguintes:

- Novos projetos atendendo às necessidades de negócios.
- Novos projetos entregues no prazo e no custo previstos.
- Novos sistemas implementados apropriadamente.
- As alterações não afetando as operações de negócios atuais.

Figura 8 Domínios do CobiT no Framework



Fonte: ITGI (2007)

Assim, segundo o ITGI (2007), os benefícios de implementar este framework como modelo de governança incluem o melhor alinhamento ao negócio, uma clara visão para os executivos sobre o que a TI faz, uma melhor divisão das responsabilidades baseada nos processos, aceitação geral por terceiros e órgãos reguladores, melhor compreensão entre todas as partes interessadas e cumprimento dos requisitos do COSO para controle do ambiente de TI.

2.1 PROCESSO DE GERÊNCIA DE MUDANÇAS

O ITGI (2007) cita no processo Gerenciar Mudanças(AI6) no CobiT, que todas as mudanças, incluindo as manutenções e correções emergenciais, relacionadas com a infraestrutura e as aplicações no ambiente de produção são formalmente gerenciadas de maneira controlada. Ainda complementa que as mudanças em procedimentos, processos, parâmetros de sistemas ou de serviços devem ser registradas, avaliadas e autorizadas antes da implementação e revisadas em seguida, tomando como base os resultados efetivos e planejados. Dessa forma, busca-se a redução de riscos de impactos negativos na estabilidade ou na integridade do ambiente de produção.

Esse processo procura fazer o controle da avaliação de impacto, minimizando erros devido a especificações de requisitos incompletas e interromper a implementação de mudanças não autorizadas.

No CobiT 4.1, para conseguir alcançar os benefícios desse processo deve-se:

- Definir e comunicar os procedimentos de mudanças, incluindo mudanças emergenciais;
- Avaliar, priorizar e autorizar mudanças;
- Acompanhar status e apresentar relatório de mudanças.

O processo AI6 pode ser medido por:

- *Quantidade de paradas ou erros em dados devido a especificações inadequadas ou avaliações de impacto críticas incompletas*
- *Retrabalho de infraestrutura ou aplicação causado por especificações de mudança inadequadas*
- *Percentual de mudanças que seguem o processo formal de controle de mudanças. (ITGI, 2007)*

O CobIT possui a tabela RACI que contem os seguintes papéis e suas responsabilidades baseados nas funções existentes deste framework, conforme segue a figura abaixo:

Figura 9. Tabela RACI

Tabela RACI	Funções										
	CEO	CFO	Executivo de Negócio	CIO	Proprietário do Processo de Negócio	Responsável por Operações	Responsável por Arquitetura	Responsável por Desenvolvimento	PMO	Conformidade, auditoria, risco e segurança	
Desenvolver e implementar um processo para registrar, avaliar e priorizar de forma consistente as solicitações de mudança;				A	I	R	C	R	C	C	C
Avaliar criticamente o impacto e priorizar mudanças baseadas em necessidades do negócio;				I	R	A/R	C	R	C	R	C
Assegurar que qualquer mudança crítica e emergencial siga o processo aprovado;				I	I	A/R	I	R			C
Autorizar mudanças;				I	C	A/R		R			
Gerenciar e disseminar informações relevantes relacionadas a mudanças				A	I	R	C	R	I	R	C

Uma tabela RACI identifica quem é responsável (R), responsabilizado (A), consultado (C) e/ou informado

Fonte: ITGI (2007)

Esta tabela ilustra cada papel e suas responsabilidades sobre cada atividade no processo de Gerência de Mudanças, sendo que:

- (R) – responsável pelo processo
- (A)- Responsabilizado pelo processo
- (C)- Consultado no processo
- (I) – Informado no processo

Segundo o CobIT, a definição para estes termos é que Responsabilizado quer dizer que “a responsabilidade é deste indivíduo” – esta é o papel que dá orientações e autoriza uma atividade. A responsabilidade é atribuída à pessoa que faz com que a tarefa seja executada. Os outros dois papéis (consultado e informado) serve para garantir de que todos que precisam, serão envolvidos e suportam o processo.

Durante o desenvolvimento desta pesquisa, o Cobit 5 foi publicado, houve algumas evoluções como a questão dos Enablers, mas não houve alteração no processo de Gerência de Mudanças.

3 PROCESSO E SUAS FORMAS DE PUBLICAÇÃO

Este capítulo descreve a definição de processo e cita algumas das formas de publicação de processos.

Um processo, na visão de Hammer e Champy (1993), é uma coleção de atividades que possui um ou mais tipos de entradas e cria uma saída que possui valor ao cliente. Um processo de negócio tem um objetivo e é afetado por eventos que ocorrem no mundo externo e em outros processos.

A documentação de um processo, segundo Sommerville (2001) é criada para que o desenvolvimento de sistemas seja gerenciado. E o autor complementa que a única maneira de tornar visível o gerenciamento de um processo de desenvolvimento de software, devido a sua característica intangível, é através da documentação do processo.

Chiavenato (2010) cita em seu livro as formas de documentação de um processo como os Fluxogramas que são gráficos que representam o fluxo ou a sequencia de atividades ou de rotinas, os Manuais que registram de forma descritiva contendo informações a respeito de procedimentos, instruções, normas de serviço, etc.

Neste trabalho, houve a utilização da ferramenta Eclipse Process Framework para o desenvolvimento do processo de Gerência de Mudanças. A descrição desta ferramenta segue no próximo capítulo.

3.1 ECLIPSE PROCESS FRAMEWORK COMPOSER

O Eclipse Process Framework (EPF), segundo Haumer (2007), é uma ferramenta para engenheiros de processos, líderes de projetos e gerente de projetos que são responsáveis por manter e implementar em organizações ou em projetos individuais. Possui dois objetivos principais:

1. Prover aos envolvidos no desenvolvimento uma base de conhecimento que os permitem explorar, gerenciar e construir conteúdo.
2. Prover capacidades de engenharia de processos no apoio aos engenheiros de processos e gerentes de projetos na seleção, e também uma montagem rápida de processos para desenvolvimento de seus projetos concretos.

O EPF, ainda segundo este mesmo autor, provê soluções para problemas comuns que equipes de desenvolvimento enfrentam quando estão adquirindo ou gerenciando seus métodos e processos como:

- Equipes de desenvolvimento precisam de acesso fácil e centralizado à informação;
- Dificuldade de integrar processos de desenvolvimento que possuem seus próprios formatos diferentes.
- Equipes precisam de uma base de conhecimento atualizada para auto aprendizado em métodos e melhores práticas;

- Equipes precisam de apoio para estimar o tamanho de seus processos;
- Garantir observância a práticas padronizadas;
- Execução com eficácia de processos em projetos

O EPF é constituído de alguns elementos como:

- Conteúdo de Métodos – descreve o que deve ser produzido, as habilidades necessárias requeridas e a explicação passo-a-passo descrevendo como um objetivo de desenvolvimento específico é atingido.
- Processos – definem como o trabalho está sendo realizado pelos autores e como os produtos de trabalhos estão sendo produzidos durante um período. Relaciona os elementos do conteúdo de métodos e os colocam em sequencias pré-ordenadas que são personalizadas para tipos específicos de projetos.

4 DESENVOLVIMENTO

Tendo em vista o tempo médio de desenvolvimento de um sistema, não foi possível realizar um estudo de caso para verificação deste modelo proposto. A justificativa deve-se ao fato de que um desenvolvimento de sistemas leva em média cerca de 1 ano e meio a 2 anos, sendo que o curso de pós-graduação *lato sensu* tem duração de 1 ano e meio. Exemplo disso, tem-se o Sistema de Atendimento à Mulher no órgão da Secretaria de Políticas para as Mulheres. Este levou cerca de 1 ano e 9 meses para sua construção, e no qual seria inviável a implementação desta proposta de modelo para Gerência de Mudanças.

Outro fator que dificulta a implementação de um estudo de caso é a complexidade de coletar amostras de software desenvolvidas que tenham executados integralmente o Open Up. Elimina-se assim, variáveis que influenciam a pesquisa na região onde está sendo desenvolvida esta proposta de modelo.

O desenvolvimento deste modelo é baseado em características objetivas de modelos já consagrados no mercado como o Open Up e CobIT e na inter-relação entre seus processos.

Primeiramente, verificou-se que o Open Up não possui um processo formal de Gerência de Mudanças. Para a criação deste processo observou-se as atividades da tabela RACI do CobIT conforme figura 7. Cada atividade possui seus atores com suas responsabilidades definidas. Desta forma, o modelo criado aproveita-se os atores, respeitando a equivalência destes, e atividades do CobIT e os inclui no processo de Gerência de Mudanças do Open Up.

Montou-se então, com as atividades da tabela RACI, um fluxo do processo de Gerência de Mudanças.

4.1 PROPOSTA DE UM MODELO EM FORMA DE PLUGIN DE GERÊNCIA DE MUDANÇAS EM ATENDIMENTO AO COBIT

O objetivo é a criação de um processo formal para o Open Up que atenda ao processo de Gerencia de Mudanças do CobiT 4.1 no desenvolvimento de sistemas com o modelo OpenUp. Assim, criou-se um plugin no Eclipse Process Framework (EPF) em conformidade com o A16 do CobiT 4.1.

O Eclipse Process Framework é um software de criação de processos customizáveis, abrangendo uma grande variedade de tipos de projetos e estilos de desenvolvimento.

A primeira etapa foi a criação dos papéis e suas responsabilidades baseados nas funções existentes da tabela RACI do Cobit, conforme a figura 7 no capítulo 3.1.

Esta tabela ilustra cada papel e suas responsabilidades sobre cada atividade no processo de Gerência de Mudanças, sendo definidos também quem é (R) responsável, (A) responsabilizado, (C) consultado e (I) informado no processo.

Segundo o CobiT, o processo de gerenciamento de mudanças não possui atividades para o CEO, CFO e Executivo.

O Eclipse Process Framework apresenta duas categorias de papéis diferentes a serem implementados. Dessa forma, adaptamos as categorias de responsabilidades do CobiT ao EPF da seguinte maneira:

- Executores Primários como (R)responsáveis pelo processo;
- Executores Adicionais como (A)responsabilizado, (C)consultado e (I)informado no processo.

Para que o modelo Open UP esteja em conformidade com o AI6, segue a comparação de papéis existentes segundo suas compatibilidades no Open UP x CobiT.

Quadro 2 – Comparação de papéis Open UP x CobiT

Open UP	CobiT
Analista	
Arquiteto	Responsável por Arquitetura
Desenvolvedor	Responsável por Desenvolvimento
Gerente de projeto	PMO
Stakeholder	Proprietário do Processo de Negócio
Testador	
Qualquer papel	
	Responsável por Operações
	CIO
	Conformidade Auditoria Risco e Segurança
	Responsável pela Administração de TI

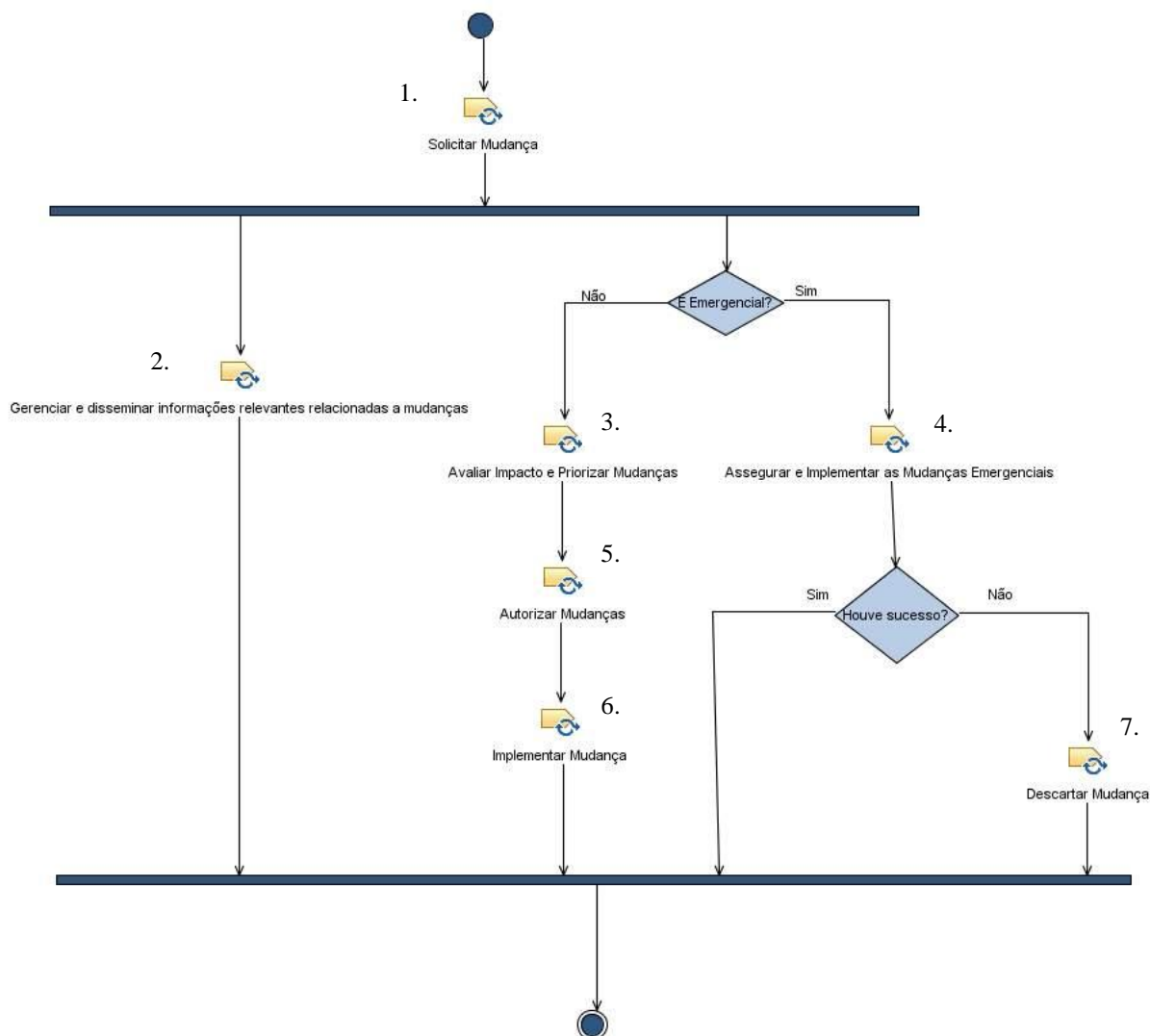
Fonte: Produzido pelo autor do trabalho

Como mostra a tabela acima, o Open Up possui papéis semelhantes ao do CobiT, assim como não possui alguns papéis que existem no CobiT.

Assim, para o OpenUP atender aos requisitos do A16 do CobiT, aproveitaremos os papéis semelhantes e adicionaremos ao plugin criado os papéis do CobiT que não estão presentes no Open UP. E ainda, vale dizer que qualquer papel pode fazer uma registrar uma solicitação de mudança.

Com os papéis definidos, criamos agora as atividades a serem realizadas no âmbito da Gerencia de Mudanças do CobiT, onde cada uma delas atende a um objetivo de controle do A16. Então, montamos assim um processo de gerencia de mudanças na figura abaixo:

Figura 10 Processo Formal do Fluxo de Gerência de Mudanças



Fonte: Produzido pelo autor do trabalho

Onde:

1 - Solicitar Mudança - proveniente do Open UP, onde se dá o início de uma solicitação de mudança.

2 - Gerenciar e disseminar informações relevantes relacionadas a mudanças - Esta atividade visa a estabelecer um sistema de acompanhamento e relatórios de mudanças e é executada concorrentemente em todo o processo de mudança.

3 - Avaliar Impacto e Priorizar Mudanças - Avaliar criticamente o impacto e priorizar mudanças baseadas em necessidades do negócio;

4 - Assegurar e Implementar as Mudanças Emergenciais- Assegurar que qualquer mudança crítica e emergencial siga o processo aprovado; Uma mudança será emergencial somente se o CIO a considerar desta forma.

5 - Autorizar Mudanças - Esta atividade constitui na tarefa de autorizar mudanças.

6 - Implementar Mudança - Atividade onde executa a implementação da mudança solicitada.

7 – Descartar Mudança – Atividade onde se descarta a mudança implementada que não foi bem sucedida.

O quadro abaixo demonstra a relação dos objetivos de controle do CobiT com as respectivas atividades.

Quadro 3 – Relação dos Objetivos de Controle do CobIT com Atividades de Gerência de Mudança

OBJETIVOS DE CONTROLE DO COBIT	ATIVIDADES/TAREFAS
AI6.1 Padrões e Procedimentos de Mudança Estabelecer procedimentos formais de gerenciamento de mudanças para lidar de modo padronizado com todas as solicitações de mudança em aplicações, procedimentos, processos, parâmetros de sistema, parâmetros de serviço e plataformas subjacentes (inclusive solicitações de manutenção e reparo).	É atendida pela criação do plugin de Gerencia de Mudanças
AI6.2 Avaliação de Impacto, Priorização e Autorização Avaliar todas as solicitações de mudança de modo estruturado com relação a impactos no sistema operacional e na respectiva funcionalidade. Assegurar que todas as mudanças sejam categorizadas, priorizadas e autorizadas.	Avaliar Impacto e Priorizar Mudanças Autorizar Mudanças
AI6.3 Mudanças de Emergência Estabelecer um processo para definição, solicitação, testes, documentação, avaliação e autorização de mudanças de emergência que não sigam o processo de mudança estabelecido.	Assegurar e Implementar as Mudanças Emergenciais
AI6.4 Acompanhamento de Status e Relatórios de Mudanças Estabelecer um sistema de acompanhamento e relatórios de mudanças para documentar mudanças rejeitadas, comunicar o status	Gerenciar e disseminar informações relevantes relacionadas a mudanças

de mudanças aprovadas e em andamento e executar mudanças. Garantir que as mudanças autorizadas sejam implementadas conforme planejado.	Implementar Mudança
AI6.5 Finalização da Mudança e Documentação Atualizar a documentação os procedimentos do sistema e de usuários sempre que forem implementadas mudanças no sistema.	Gerenciar e disseminar informações relevantes relacionadas a mudanças

Fonte: Produzido pelo autor do trabalho

Quadro 4 - Relação de Atividades x Papéis

	CEO	CFO	Executivo de Negócio	CIO	Proprietário do Processo de Negócio	Responsável por Operações	Responsável por Arquitetura	Responsável por Desenvolvimento	Responsável pela Administração de TI	PMO	Conformidade, auditoria, risco e segurança
Solicitar Mudança	P	P	P	P	P	P	P	P	P	P	P
Avaliar Impacto e Priorizar Mudanças				A	P	P	A	P	A	P	A
Assegurar e Implementar as Mudanças Emergenciais				A	A	P	A	P			A
Autorizar Mudanças				A	A	P		P			
Gerenciar e disseminar informações relevantes relacionadas a mudanças				A	A	P	A	P	A	P	A
Implementar Mudança						P		P			

(P) Executor Primário, (A) Executor Adicional

Fonte: Produzido pelo autor do trabalho

Assim, o quadro 4 mostra uma relação do quadro 2 e quadro 3, mostrando assim quem desempenhará nas atividades do processo de Gerência de Mudanças e qual será sua função nessa tarefa.

Vale ressaltar que, apesar de ter surgido o Cobit 5 no período da realização dessa pesquisa, o processo AI6 do CobiT 4.1 continua inalterado em comparação com o Cobit 5, mudando apenas seu nome para BAI 6 (Build, Acquire, Implement) e mantendo as práticas de gerenciamento como citados abaixo:

BAI06.01 Avaliar, priorizar e autorizar solicitações de mudanças

Avaliar todas solicitações de mudanças para determinar o impacto nos processos de negócios e nos serviços de TI e para avaliar se a mudança irá afetar de forma adversa o ambiente operacional e introduzir um risco inaceitável. Garantir que mudanças sejam registradas, priorizadas, categorizadas, avaliadas, autorizadas, planejadas e programadas.

BAI06.02 Gerenciar mudanças emergenciais.

Gerenciar cuidadosamente mudanças emergenciais para minimizar maiores incidentes e garantir que a mudança será controlada e implementada de forma segura. Verificar que as mudanças de emergência serão avaliadas e priorizadas de forma apropriada depois da mudança.

BAI06.03 Acompanhar e reportar status de mudanças.

Manter um sistema de acompanhamento e de relatórios para documentar mudanças rejeitadas, comunicar o estado de mudanças aprovadas, mudanças em processo e mudanças completadas. Garantir que mudanças aprovadas serão implementadas conforme planejado.

BAI06.04 Finalizar e documentar as mudanças

Sempre que uma mudança for implementada, atualizar adequadamente a documentação de soluções e os procedimentos afetados pela mudança.

(Cobit 5)

Nota-se que os objetivos de controle são os mesmos, apenas excluindo o AI6.1 Padrões e Procedimentos de Mudança, mas o qual não afeta o desenvolvimento deste trabalho por tratar-se da criação de um processo formal o qual já será atendido pela criação deste plugin de Gerência de Mudanças.

Assim, com a união de todos estes elementos criados no plugin pelo software EPF, surge então a proposta do processo formal de Gerência de Mudanças para contribuir com o modelo de desenvolvimento de sistemas Open Up. Dessa forma, este modelo terá este processo formalizado em conformidade com o framework CobiT.

CONCLUSÃO

Conclui-se neste trabalho que houve a possibilidade de adaptar uma disciplina do processo iterativo de desenvolvimento de software Open Up para atender aos requisitos dos objetivos de controle do processo de Gerência de Mudanças do modelo de governança de TI CobIT e também criar um processo formal de Gerência de Mudanças para ser utilizado na construção ou no desenvolvimento softwares.

Esta adaptação foi possível devido ao relacionamento de papéis e suas responsabilidades entre o Open Up e o CobIT , e também com a atribuição de atividades a serem realizadas por estes papéis para a formalização do processo de Gerência de Mudanças. Atividades estas que foram levantadas em conformidade com os requisitos do framework CobIT.

Realizando-se estas mudanças de maneira formalizada e padronizada, tende-se a reduzir o retrabalho e atrasos nas entregas de sistemas sendo desenvolvidos na corporação.

Para trabalhos futuros, sugere-se a possibilidade de implementar novas integrações de outras áreas de processos de desenvolvimento de softwares com outros processos de modelos de governança de TI.

REFERÊNCIAS

- AGILE MANIFESTO. *Agile Manifesto*. Disponível em: <http://agilemanifesto.org/>
Acesso em: 05 jan. 2013
- ALTER, S. *Information systems: a management perspective*. 3. ed. Indiana, EUA: Addison Weley, 1999.
- BALDUINO, R. *Introduction to OpenUP*. Disponível em: <http://www.eclipse.org/epf/general/OpenUP.pdf>. Acesso em: 14 jan. 2013.
- BOEHM, B. A View of 20th and 21st Century Software Engineering. *Proceedings of the 28th international*, Nova York, EUA, 2006 pp. 12-29. Disponível em: <http://dl.acm.org/citation.cfm?id=1134288> Acesso em: 15 de dez. 2012.
- BROOKS, F. P. No Silver Bullet: Essence and Accidents of Software Engineering. *IEEE Computer*, n. 20,p.10-19, abr.1987. Disponível em: <http://faculty.salisbury.edu/~xswang/Research/Papers/SERelated/no-silver-bullet.pdf>
Acesso em: 05 de jan. 2013
- CHIAVENATO, I. *Introdução à teoria geral da administração: uma abordagem abrangente da moderna administração das organizações*. 7. ed. Rio de Janeiro: Elsevier, 2003.
- CHIAVENATO, I. *Os novos paradigmas: Como as mudanças estão mexendo com as empresas* 5. ed. Barueri, SP: Manole,2008.
- CHIAVENATO, I. *Iniciação a sistemas, organização e métodos SO&M*. SP: Manole,2010.
- FERNANDES, A. A. *Implantando a Governança de TI*. Rio de Janeiro: Brasport,2008.
- HAMMER, M.; CHAMPY, J. *Reengineering the Corporation: A Manifesto for Business Revolution*. New York, NY: Harper Business, 1993.
- HAUMER, P. *Eclipse Process Framework Composer*. Disponível em: <http://www.eclipse.org/epf/general/EPFComposerOverviewPart1.pdf> Acesso em: 15 de abr. 2013
- HELDMAN, K. *Gerência de Projetos*. Rio de Janeiro, RJ: Elsevier,2006.
- IT GOVERNANCE INSTITUTE. *CobIT 4.1 Information Systems Audit and Control Association, Control Objectives for Information and Related Technology*. Rolling Meadows, IL,EUA: 2007.
- O'BRIEN, J. A. *Sistemas de informação e as decisões gerenciais na era internet*. 2. ed. São Paulo, SP: Saraiva,2004.
- OLIVEIRA, D. d. *Sistemas, organização e métodos: uma abordagem gerencial*. 17. ed. São Paulo, SP: Atlas,2007.
- PRESSMAN, R. *Engenharia de Software*. São Paulo: McGraw-Hill Interamericana do Brasil, 2001.
- PRESSMAN, R. S. *Engenharia de Software*. São Paulo: Pearson,2007.
- ROYCE, W. *Software Project Management - A unified framework*. EUA: Addison Wesley Longman, 1998.

SCHWABER, K.; BEEDLE, M. *Agile Software Development with SCRUM*. Pennsylvania, USA: Prentice-Hall, 2002.

SENGE, P. *A dança das mudanças: os desafios de manter o crescimento e o sucesso em organizações que aprendem*. Rio de Janeiro: Campus, 2000.

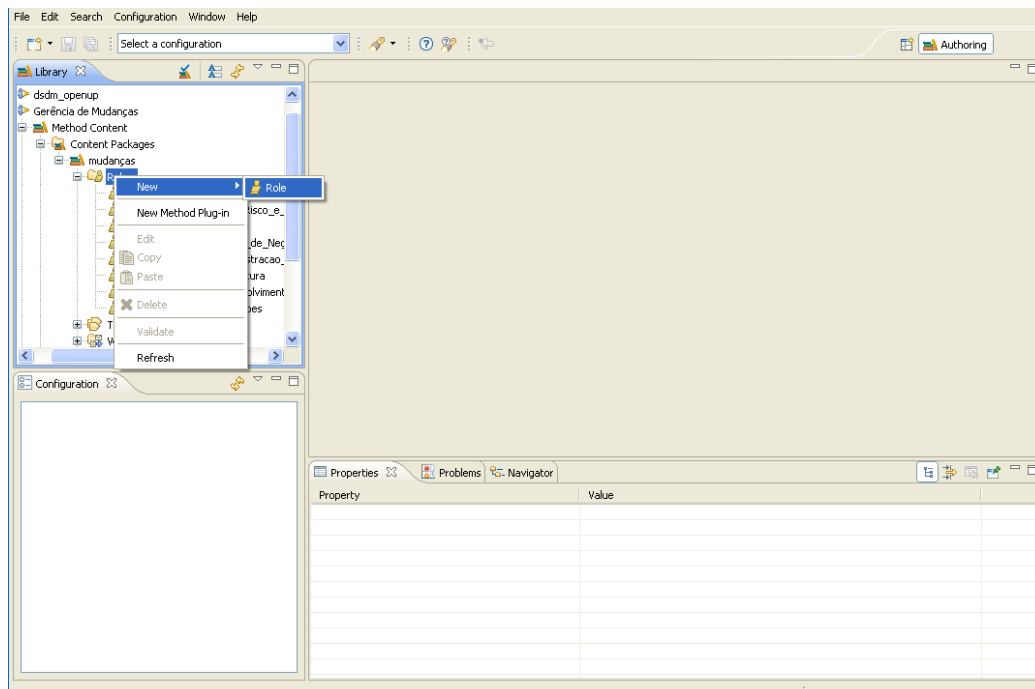
SOMMERVILLE, I. *Software Engineering*. Harlow, Reino Unido: Pearson Education, 2001.

SZALVAY, V. *An Introduction to Agile Software Development*. Danube Technologies. EUA: Collabnet, 2004

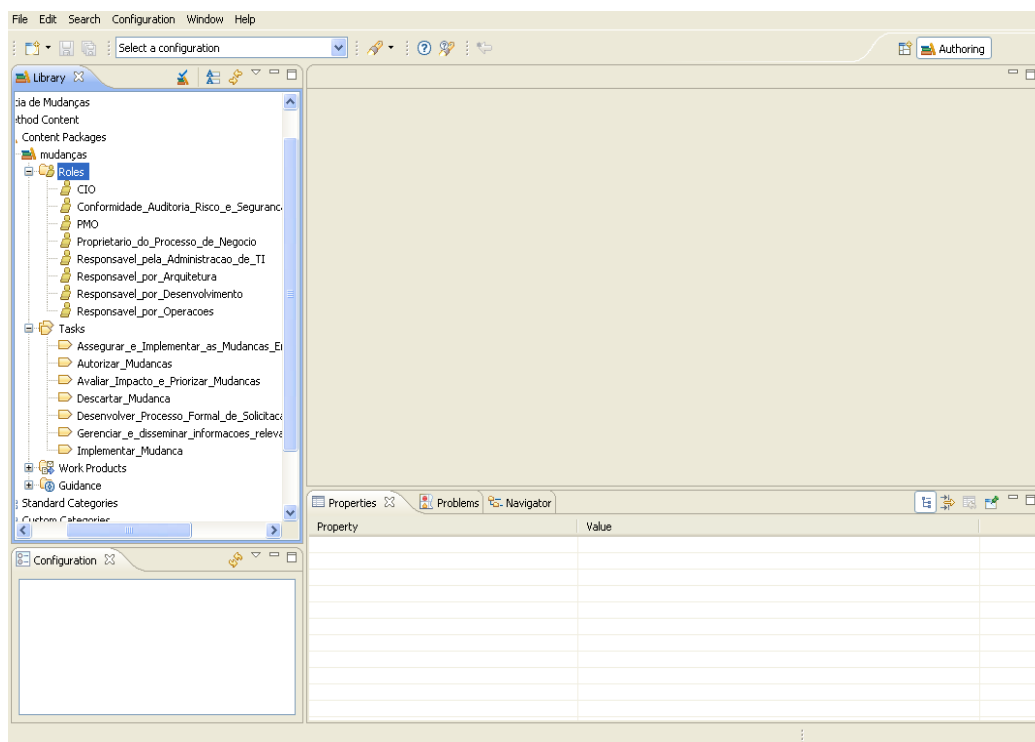
THE STANDISH GROUP INTERNATIONAL. *Extreme Chaos*. Dennis, MA, EUA: The Standish Group International, 2009.

WHITTEN, N. *Managing software development projects - formula for success*. 2.ed. NY, EUA: John Wiley & Sons, 1995

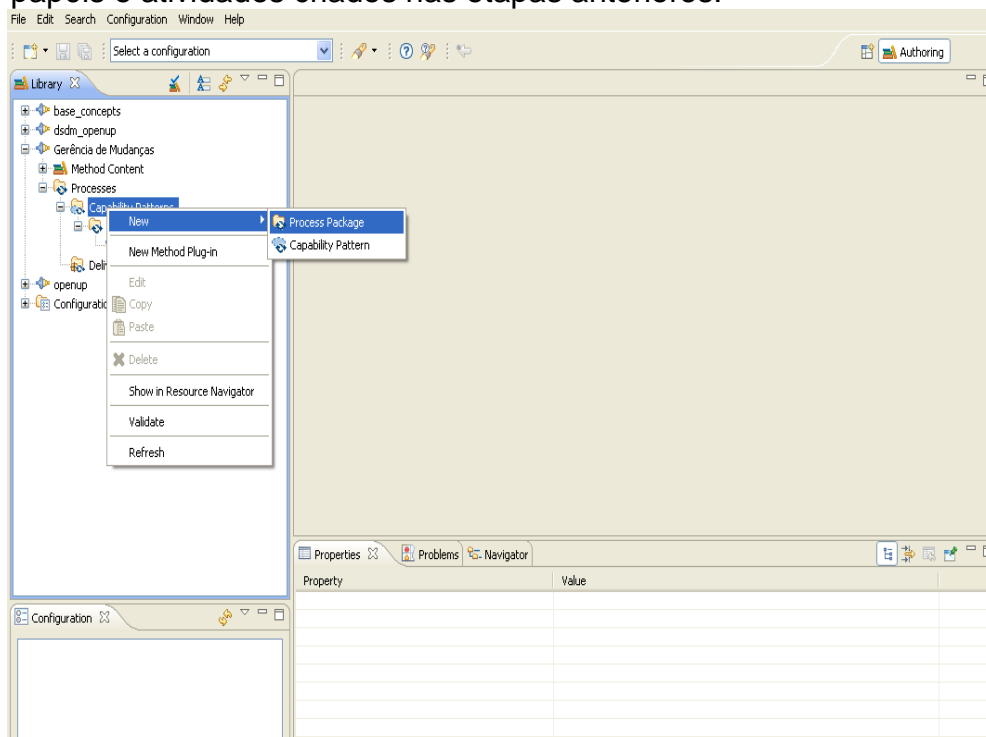
- 4- Na etapa de agora, deve-se criar os papéis e atividades do COBIT conforme a tela abaixo:



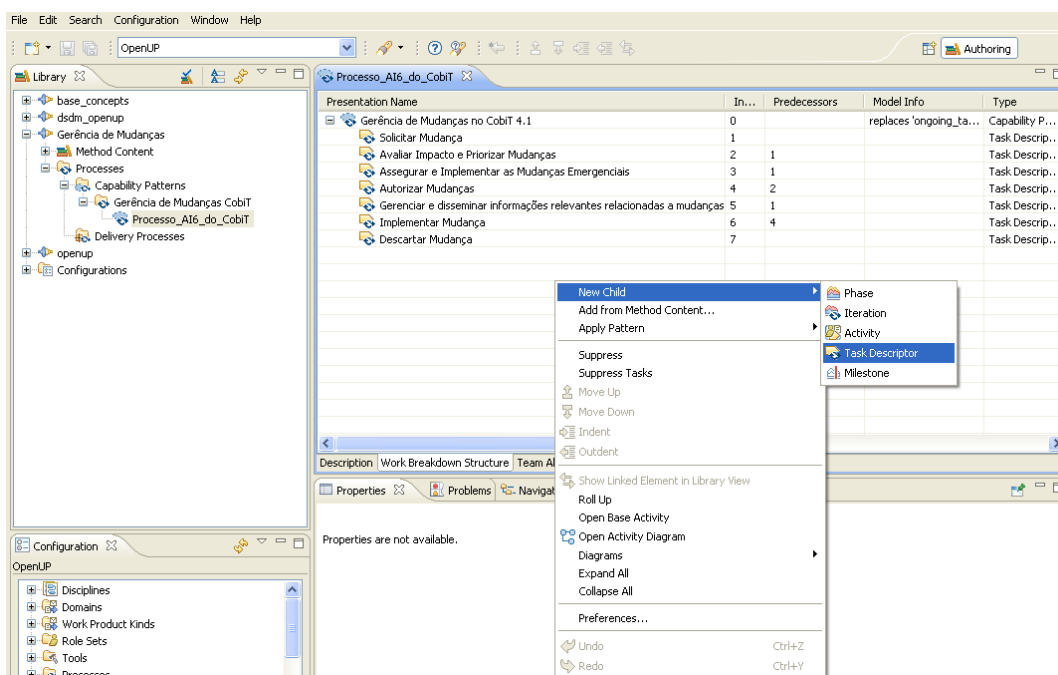
5. A tela abaixo mostra como ficou após a criação dos papéis e das atividades do COBIT para o A16 Gerência de Mudanças.



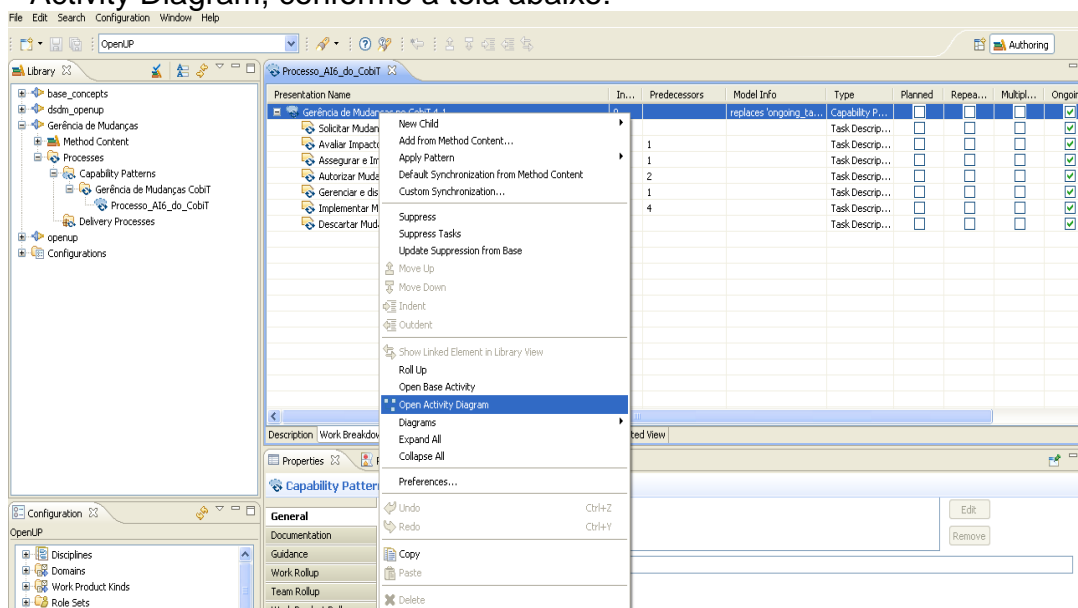
- 5- Nesta etapa, deve-se criar o processo de Gerência de Mudanças com os papéis e atividades criados nas etapas anteriores.



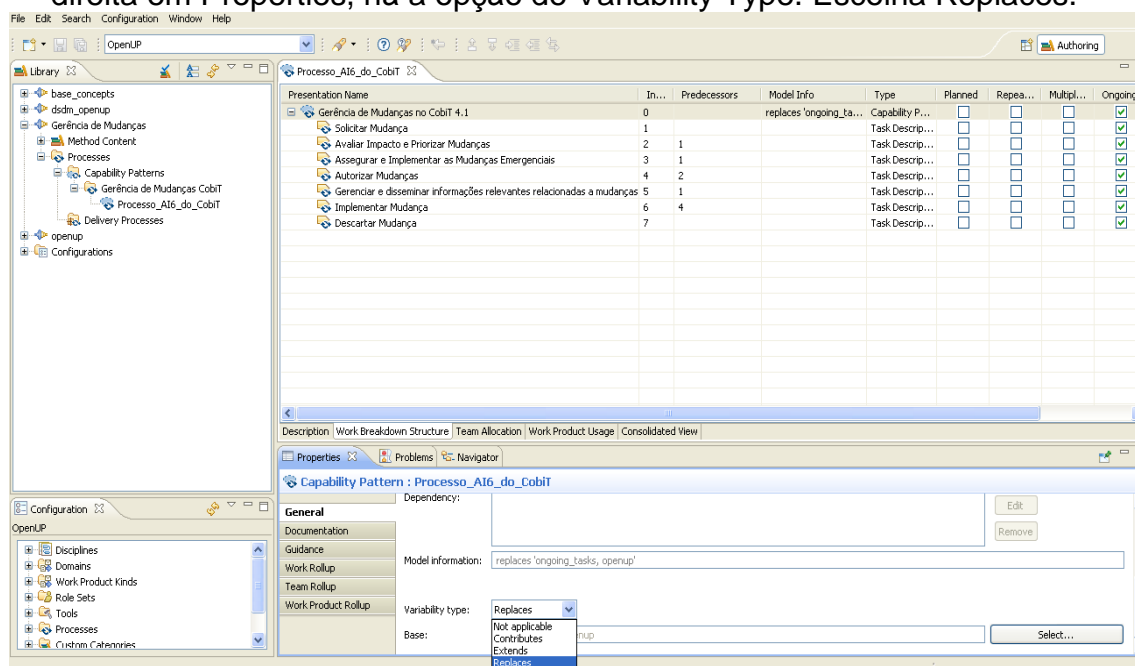
- 6- Após isso, clique com botão direito no Process Package que criou, e clique em New, Capability Pattern e nomeie como Processo_Al6_do_Cobit. Escolhe Open UP como Default Configuration.
- 7- Agora abra o Processo criado, e na tela a direita vá até Work Breakdown Structure, clique com botão direito em cima do processo e faça como a tela abaixo até criar todas as atividades.



- 8- Agora deve-se referenciar estas atividades com as atividades criadas em Tasks dentro de Method Content. Basta clicar em uma atividade criada e ir em Properties na tela abaixo. Clique em Link Method Content e escolha a atividade a ser referenciada.
- 9- Ainda em Properties, há a opção Dependency, onde se ordena as atividades a serem realizadas.
- 10- Após ordenadas as atividades, pode-se fazer o desenho do processo com mais detalhes. Para isso, clique com botão direito no processo e escolha Open Activity Diagram, conforme a tela abaixo:



- 11- Terminando estas etapas, agora pode-se substituir este processo por um processo existente no Open UP. Abra o processo criado, e abaixo na tela à direita em Properties, há a opção de Variability Type. Escolha Replaces.



- 12- O processo já pode ser publicado através do menu de ferramentas, Configuration, Publish.